Relational Databases with MySQL Week 10 Assignment

Points possible: 70

Livermore, Tyler

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document to the repository. Additionally, push an .sql file with all your queries and your Java project code to the same repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

In this week's coding activity, you will create a menu driven application backed by a MySQL database.

To start, choose one item that you like. It could be vehicles, sports, foods, etc....

Create a new Java project in Eclipse.

Create a SQL script in the project to create a database with one table. The table should be the item you picked.

Write a Java menu driven application that allows you to perform all four CRUD operations on your table.

Tips:

The application does not need to be as complex as the example in the video curriculum.

You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

Remember that PreparedStatment.executeQuery() is only for Reading data and .executeUpdate() is used for Creating, Updating, and Deleting data.

Remember that both parameters on PreparedStatements and the ResultSet columns are based on indexes that start with 1, not 0.

Screenshots of Code:

- MySQL:

```
1  create database if not exists npb;
2
3  use npb;
4
5  drop table if exists players;
6
7  create table players (
8      id int(3) not null auto_increment,
9      lastname varchar(30) not null,
10     firstname varchar(30) not null,
11     team varchar(30) not null,
12     pos varchar(3) not null,
13     primary key(id)
14 );
```

```
mysql> use npb
Database changed
mysql> desc players;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| id        | int         | NO   | PRI | NULL    | auto_increment |
| lastname  | varchar(30) | NO   |     | NULL    |                |
| firstname | varchar(30) | NO   |     | NULL    |                |
| team      | varchar(30) | NO   |     | NULL    |                |
| pos       | varchar(3)  | NO   |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

- Java:

Application.java | Menu.java

```java
package week10Project;

public class Application {

    public static void main(String[] args) {
        Menu menu = new Menu();
        menu.start();
    }

}
```

Application.java | Menu.java

```java
package week10Project;

import java.sql.*;

public class Menu {

    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList(
            "Display Players",
            "Create Player",
            "Update Player",
            "Delete Player");

    public void start() {
        String selection = "";

        do {
            printMenu();
            selection = scanner.nextLine();

            if (selection.equals("1")) {
                displayPlayers();
            } else if (selection.equals("2")) {
                createPlayer();
            } else if (selection.equals("3")) {
                updatePlayer();
            } else if (selection.equals("4")) {
                deletePlayer();
            }

            System.out.println("Press enter to continue.");
            scanner.nextLine();
        } while (!selection.equals("e"));
        System.out.println("Goodbye.");
    }

    private void printMenu() {
        System.out.println("Select an option:\n-------------------------");
        for (int i = 0; i < options.size(); i++) {
            System.out.println(i + 1 + ":" + options.get(i));
        }
    }
```

Application.java | *Menu.java

```
46
47      private void displayPlayers() {
48          try {
49              Class.forName("com.mysql.cj.jdbc.Driver");
50              Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/npb",
51                      "root",
52                      "password"); //change to your MySQL password
53              System.out.println("Connection established.");
54
55              Statement stmt = con.createStatement();
56              ResultSet rs = stmt.executeQuery("Select * from players");
57              while (rs.next()) {
58                  System.out.println("Player ID: [" + rs.getInt(1) + "]");
59                  System.out.println("Player: " + rs.getString(2) + ", " + rs.getString(3));
60                  System.out.println("Team: " + rs.getString(4));
61                  System.out.println("Position: " + rs.getString(5));
62                  System.out.println("----------------------------------------");
63              }
64
65          } catch (Exception e) {
66              System.out.println(e);
67          }
68      }
69
70      private void createPlayer() {
71          try {
72              Class.forName("com.mysql.cj.jdbc.Driver");
73              Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/npb",
74                      "root",
75                      "password"); //change to your MySQL password
76              System.out.println("Connection established.");
77
78              PreparedStatement stmt = con.prepareStatement(
79                      "Insert Into players(id, lastname, firstname, team, pos) VALUES (?, ?, ?, ?, ?);"
80                      );
81              stmt.setInt(1, 5);
82              stmt.setString(2, "Laird");
83              stmt.setString(3, "Brandon");
84              stmt.setString(4, "Marines");
85              stmt.setString(5, "DH");
86              int i = stmt.executeUpdate();
87
88              System.out.println(i + " records affected.");
89
90          } catch (Exception e) {
91              System.out.println(e);
92          }
93      }
94
```

```
94
95      private void updatePlayer() {
96          try {
97              Class.forName("com.mysql.cj.jdbc.Driver");
98              Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/npb",
99                      "root",
100                     "password"); //change to your MySQL password
101             System.out.println("Connection established.");
102
103             PreparedStatement stmt = con.prepareStatement(
104                     "Update players set pos = ? where lastname = ?;");
105             stmt.setString(1, "3B");
106             stmt.setString(2, "Kurebayashi");
107             int i = stmt.executeUpdate();
108
109             System.out.println(i + " records affected.");
110
111         } catch (Exception e) {
112             System.out.println(e);
113         }
114     }
115
116     private void deletePlayer() {
117         try {
118             Class.forName("com.mysql.cj.jdbc.Driver");
119             Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/npb",
120                     "root",
121                     "password"); //change to your MySQL password
122             System.out.println("Connection established.");
123
124             PreparedStatement stmt = con.prepareStatement(
125                     "Delete from players where id = ?;"
126                     );
127             stmt.setInt(1, 5);
128             int i = stmt.executeUpdate();
129
130             System.out.println(i + " records affected.");
131
132         } catch (Exception e) {
133             System.out.println(e);
134         }
135     }
136 }
```

Screenshots of Running Application:

- Show players and exit menu:

```
Select an option:
-------------------------
1:Display Players
2:Create Player
3:Update Player
4:Delete Player
1
Connection established.
Player ID: [1]
Player: Tongu, Yuma
Team: Buffaloes
Position: C
-----------------------------------------
Player ID: [2]
Player: Ishikawa, Ayuma
Team: Marines
Position: P
-----------------------------------------
Player ID: [3]
Player: Kondoh, Kensuke
Team: Fighters
Position: CF
-----------------------------------------
Press enter to continue.

Select an option:
-------------------------
1:Display Players
2:Create Player
3:Update Player
4:Delete Player
e
Press enter to continue.

Goodbye.
```

- Adding a player:

```
Select an option:
-------------------------
1:Display Players
2:Create Player
3:Update Player
4:Delete Player
2
Connection established.
1 records affected.
Press enter to continue.
```

```
1
Connection established.
Player ID: [1]
Player: Tongu, Yuma
Team: Buffaloes
Position: C
-----------------------------------------
Player ID: [2]
Player: Ishikawa, Ayuma
Team: Marines
Position: P
-----------------------------------------
Player ID: [3]
Player: Kondoh, Kensuke
Team: Fighters
Position: CF
-----------------------------------------
Player ID: [4]
Player: Kurebayashi, Kotaro
Team: Buffaloes
Position: SS
-----------------------------------------
Player ID: [5]
Player: Laird, Brandon
Team: Marines
Position: DH
-----------------------------------------
Press enter to continue.
```

Deleting the player:

```
4:Delete Player
4
Connection established.
1 records affected.
Press enter to continue.

Select an option:
-------------------------
1:Display Players
2:Create Player
3:Update Player
4:Delete Player
1
Connection established.
Player ID: [1]
Player: Tongu, Yuma
Team: Buffaloes
Position: C
-----------------------------------------
Player ID: [2]
Player: Ishikawa, Ayuma
Team: Marines
Position: P
-----------------------------------------
Player ID: [3]
Player: Kondoh, Kensuke
Team: Fighters
Position: CF
-----------------------------------------
Player ID: [4]
Player: Kurebayashi, Kotaro
Team: Buffaloes
Position: SS
-----------------------------------------
Press enter to continue.
```

Updating player:

```
Player ID: [4]
Player: Kurebayashi, Kotaro
Team: Buffaloes
Position: SS
-----------------------------------------
Press enter to continue.

Select an option:
------------------------
1:Display Players
2:Create Player
3:Update Player
4:Delete Player
3
Connection established.
1 records affected.
Press enter to continue.
-----------------------------------------
Player ID: [4]
Player: Kurebayashi, Kotaro
Team: Buffaloes
Position: 3B
-----------------------------------------
Press enter to continue.
```

URL to GitHub Repository:

https://github.com/tylerjlivermore/Week10_MySQL_ConnectUsingJava